Cows and Bulls - a simple game

Srini, rs@toprllc.com

May 2, 2021

1 Introduction

`https://en.wikipedia.org/wiki/Bulls_and_Cows` is the subject of this projectlet. A simple but captivating game, this has been the subject of programs and papers from the early days of computing.

In this projectlet, we develop both sides of the coin - the dealer where the computer thinks of the number while the human attempts to guess the number. Each guess is evaluated and the score announced.

In the player mode, the human thinks of the number while the computer attempts to guess by choosing different guesses.

1.1 Learning Objectives

Dynamic memory allocation and deallocation This projectlet uses a number of data structures that are allocated and deallocated during the course of execution. Instrumenting the project appropriately **gnatmem** in the case of **Ada** and perhaps a tool like **valgrind** in the case of **C or C++** can reveal some surprises.

2 Specifications

This game is implemented in both modes where the computer is the **dealer** as well as the **player** mode where the human thinks of the number and the computer attempts to guess the number. In theory the computer should be able to arrive at the correct soluion in 5 guesses or less. This project uses a naive approach to the solution which is reasonable for this project size ie 4 digits. It may be prudent to search for better alternatives if the size of the problem is larger.

2.1 Design

Dealer mode The dealer mode is of course trivial. The computer simply uses a time dependent random number to **think** of a number and it is the user who has to guess.

Player mode In the implementation at hand, a simplistic approach is taken in the **player** mode. As the user scores each **guess**, the program just eliminates the numbers from the list of potential solutions. Thus each score reduces the potential solution size - eventually leading up to just one potential (and the real) solution.

2.2 Memory Leak analysis

Enabling the instrumentation and inspecting the results - there are no surprises:

Listing 1: Memory Alloation/deallocation profile

```
Global information
------------------
   Total number of allocations        :13198
   Total number of deallocations      :4322
   Final Water Mark (non freed mem)   : 349.21 Kilobytes
   High Water Mark                    : 349.61 Kilobytes

Allocation Root # 1
-------------------
 Number of non freed allocations    :8858
 Final Water Mark (non freed mem)   : 207.61 Kilobytes
 High Water Mark                    : 207.61 Kilobytes
 Backtrace                          :
   numbers.adb:50 numbers.digitize

Allocation Root # 2
-------------------
 Number of non freed allocations    :   4
 Final Water Mark (non freed mem)   : 32 Bytes
 High Water Mark                    : 32 Bytes
 Backtrace                          :
   numbers-game.adb:37 numbers.game.score

Allocation Root # 3
-------------------
 Number of non freed allocations    :   2
 Final Water Mark (non freed mem)   : 88 Bytes
 High Water Mark                    : 88 Bytes
 Backtrace                          :
   g-comlin.adb:0 ??

...
...
...
```

3 Implementation

Implementation in **Ada** `https://gitlab.com/ada23/guess.git`